ALGORITMOS Y ESTRUCTURAS DE DATOS III - 1$^{\text{er}}$ Parcial

Fecha de examen: 02-OCT-2019

Nº Orden	Apellido y nombre	L.U.	# hojas [1]
165			8

Notas:	Ej1	Ej2	Ej3	Ej4	Ej5	Final
	B	B-	M	B	B	A

NOTA: 8

Aclaraciones: El parcial se aprueba con 3 ejercicios bien. Los ejercicios se deben entregar en **hojas separadas**. Cada hoja debe estar númerada e indicar el número de orden. El parcial dura 4 horas. El parcial es a libro cerrado.

1. Sea G un grafo y G^C su complemento.

   a) Decidir y demostrar si existe un grafo tal que G y G^C son conexos.

   b) Decidir y demostrar si existe un grafo tal que G y G^C no son conexos.

2. Un grafo se dice *libre de triángulos* si no existe un conjunto de 3 vértices que sean todos adyacentes entre si. Sea $G = (V, E)$ tal que $|V| = 2n$ y G es libre de triángulos . Demostrar que $|E| \leq n^2$.
   Nota: Si se sacan dos vértices adyacentes el grafo sigue siendo libre de triángulos.

3. Sea $S = (x_1, x_2, \ldots, x_n)$ una secuencia de n booleanos (1 o 0) y sea k un numero entre 1 y n. Supongamos que se pueden eliminar k ceros, queremos saber la longitud máxima que puede tener una cadena de 1s. Por ejemplo si $k = 2$ y la $S = 11001010001$ la respuesta es 3, mientras que si $k = 3$ la respuesta es 4.

   a) Diseñar un algoritmo basado en programación dinámica que indique la longitud más larga de una subsecuencia de 1s sacando a lo sumo k 0s de S. Debe tener complejidad temporal a lo sumo $O(nk)$.

   b) Demostrar que es correcto.

   c) Demostrar su complejidad temporal.

4. Sea $G = (V, E)$ un grafo con costos asociados a las aristas tal que **todos los costos son distintos.** Un segundo mejor árbol generador es un árbol generador tal que su costo es mínimo entre todos los árboles generadores que no son AGM.

   a) Decidir si el AGM es único. En caso afirmativo demostrar, si no dar un contra ejemplo.

   b) Decidir si el segundo mejor árbol generador es único. En caso afirmativo demostrar, si no dar un contra ejemplo.

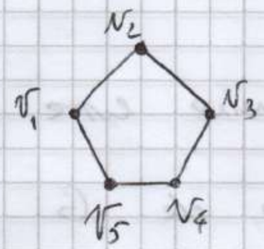
5. Oto Metedac entrega paquetes en su moto llevandolos desde su deposito a cada uno de los destinos. Todas las mañanas obtiene la lista de paquetes a entregar. Su moto le permite llevar solo un paquete por vez por lo que entre entrega y entrega debe volver al deposito. Oto quiere realizar las entregas con el menor costo (en tiempo) posible. Conocemos el mapa de la ciudad, es decir, las esquinas y las calles con su direccion de circulación. Tambien conocemos el tiempo que es necesario para recorrer cada cuadra. Queremos ayudar a Oto a conocer el tiempo que tarda en completar todos los pedidos. Modelar este problema como un problema de grafos. Decidir el algoritmo para resolver el problema y calcular su complejidad. El algoritmo debe tener complejidad temporal a lo sumo $O(n^2)$, donde n es la cantidad de esquinas.
   Nota: todos los pesos son no negativos; se puede suponer que el deposito y los lugares donde hay que dejar la mercadería estan en las esquinas.

B

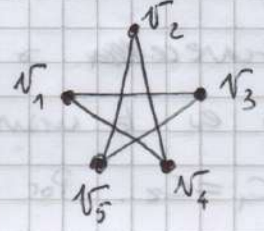
[redacted]

1
165

(1) a) Tomo G el grafo:



El complemento de G es $\overline{G}$:



Ambos grafos son conexos, ya que existe un camino entre cada par de vértices v_i, v_j con $i,j \in \{1,...,5\}$

Por lo tanto, existe al menos un grafo G tal que G y $\overline{G}$ son conexo.

b) Quiero probar que si G no es conexo, necesariamente su complemento $\overline{G}$ tiene que serlo.

Sea G cualquier grafo no conexo, tenemos que G tiene 2 o más componentes conexos.

Al complementar G, tenemos que cada vértice v de G está conectado por una arista con al menos todos los vértices de todos los componentes conexos de G salvo la suya

Si $\bar{G}$ no fuera conexo, existirían al menos dos vértices $v_i, v_j \in V(\bar{G})$ tal que no existe un camino entre ellos.

Veamos que sí existe un camino entre ellos:

Sean C_1 la componente conexa de G que contiene a v_i y C_2 la que contiene a v_j.

Si $C_1 \neq C_2$, por lo dicho anteriormente, la arista $e / v_i \overset{e}{—} v_j$ pertenece a $\bar{G}$, ya que, de no ser así, estarían en la misma componente conexa de G, o sea $C_1 = C_2$. Por lo tanto, si $C_1 \neq C_2$, tenemos que existe un camino entre v_i y v_j.

Si $C_1 = C_2$, podría pasar que la arista $f / v_i \overset{f}{—} v_j$ pertenezca a G o no.

Si pertenece, no pertenece a $\bar{G}$, pero podemos tomar un camino de v_i a un vértice cualquiera v_k de una componente conexa $C_3 \neq C_1$ y un camino de v_k a v_j por lo dicho anteriormente. Luego, un camino de v_i a v_j es $v_i — v_k — v_j$.

Si no pertenece, pertenece a $\bar{G}$, luego existe al menos un camino entre v_i, v_j que es por ejemplo la arista que los une.

Como v_i y v_j son vértices cualesquiera de $\bar{G}$, entre todo par de vértices de $\bar{G}$ existe un camino. Luego, no puede pasar que si G no es conexo, $\bar{G}$ tampoco lo sea.

2
165

(2) $G=(V,E)$ / $|V| = 2m$ y G libre de triángulos
pvq $|E| \leq m^2 = \left(\frac{|V|}{2}\right)^2$

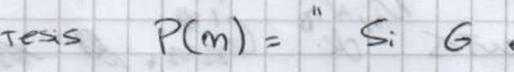
Tomo la hipótesis $P(m)$ = " Si G es un grafo libre de triángulos tal que $|V(G)| = 2m$, entonces $|E(G)| \leq m^2$"

Demostración por inducción en m:

Caso base:

$P(1)$: ~~tenemos~~ tenemos que $|V| = 2\cdot 1 = 2$ y G libre de triángulos:

Los grafos con dos vértices pueden ser



Ambos son libres de triángulos y $|V|=2$. También tenemos que $|E(G_1)| = 0 \leq 1^2$ y $|E(G_2)| = 1 \leq 1^2$. Luego, la implicación es verdadera.

Paso inductivo:

Queremos ver que $P(m) \Rightarrow P(m+1)$

Sea G' grafo libre de triángulos tal que $|V(G')| = 2(m+1)$, pvq usando la hipótesis inductiva llegamos a que $|E(G')| \leq (m+1)^2$.

Tomo G' de esa forma ya que, de no cumplir el antecedente, la implicación es verdadera y no hay nada que probar.

Como queremos usar nuestra hipótesis, eliminamos dos vértices adyacentes v_i, v_j de G' ~~...~~ y llamamos G al grafo resultante.

Notemos que, si no existieran dos vértices adyacentes, tendríamos todos nodos aislados y 0 aristas, por lo tanto tendríamos un grafo libre de triángulos y con $0 \leq m^2$. Luego, la afirmación sería verdadera. Dicho esto, podemos asumir que dichos vértices adyacentes existen. OK

Tenemos que G es libre de triángulos porque lo era G', ya que al eliminar nodos nunca se podrían generar un conjunto de 3 vértices adyacentes. Además, $|V(G)| = 2(m+1) - 2 = 2m$. G cumple nuestra hipótesis inductiva, entonces $|E(G)| \leq m^2$

Veamos que al agregar los dos vértices v_i, v_j llegamos a lo que queremos.

Tenemos que $|E(G')| = |E(G)| + d(v_i) + d(v_j) - 1$ ya que le sumamos las aristas de v_i, v_j y restamos 1 por haber sumado 2 veces la arista $v_i - v_j$ (que existe porque eran adyacentes)

queremos llegar a que $|E(G')| \leq (m+1)^2$

Tenemos que

$$|E(G')| = |E(G)| + d(v_i) + d(v_j) - 1 \underset{\substack{\text{por} \\ \text{hipótesis inductiva}}}{\leq} m^2 + d(v_i) + d(v_j) - 1 \leq \circledast$$

Ahora bien, sabemos que en todo grafo (no pseudografo) cada nodo puede tener como máximo grado $|V|-1$, pero también sabemos que no pueden

$$\cancel{\circledast \leq m^2 + |V(G')|-1 + |V(G')|-1 - 1 = m^2 + 2(m+1)-1 + 2(m+1)-1-1 = m^2 + 2m+2-1+2m+2-1-1 = m^2+4m+1}$$

haber 3 nodos adyacentes, por lo tanto, como v_i y v_j son adyacentes, no pueden ser adyacentes a otro nodo ambos a la vez.

Esto significa que como máximo $d(v_i) + d(v_j)$ puede ser $|V(G')|$, ya que, si v_i es adyacente a todos los nodos ($d(v_i) = |V(G')|-1$), entonces $d(v_j) = 1$, si v_i es adyacente a v_j y a v_k, entonces v_j puede ser adyacente como máximo a v_i y a todos los demás salvo v_k ($d(v_i) = 2$ y $d(v_j) = |V(G')|-2$) y así sucesivamente.

Dicho esto, tenemos que:

$$d(v_j) + d(v_i) \leq |V(G')|$$ (*)

$$\Rightarrow \ * \leq n^2 + |V(G')| - 1 = n^2 + 2(n+1) - 1 =$$

$$= n^2 + 2n + 1 = (n+1)^2.$$

Llegamos a que $|E(G')| \leq (n+1)^2$, que era lo que queríamos probar. □

Bien! -

(*) Esto está bien. Pero está flojo y confusamente justificado.

[redacted]

4
165

(4) a) Veamos que el AGM es único. $(W(T_1)=W(T_2))$

Supongo que existen dos AGMs T_1, T_2, con ~~...~~ $T_1 \neq T_2$. Como son distintos, difieren en al menos una arista. Ordeno las aristas de ambos de menor a mayor, obteniendo lo siguiente.

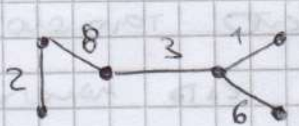
T_1	T_2
e_1	f_1
e_2	f_2
$\vdots$	$\vdots$
e_{m-1}	f_{m-1}

Busco la primera arista que difiera y sabemos que existe por hipótesis. Sean e_i y f_i las aristas diferentes, como todas las aristas tienen peso diferente, tiene que pasar $e_i > f_i$ o ~~...~~ $e_i < f_i$. Supongamos sin pérdida de generalidad ~~...~~ $e_i < f_i$. Entonces, como $e_k = f_k$ $\forall 1 \leq k \leq i$, sabemos que e_i no genera ciclos en T_2 con ~~con~~ todas las aristas menores a ella, ~~...~~ por lo tanto podemos ~~...~~ reemplazar ~~...~~ f_i por e_i y ~~...~~ eventualmente reemplazar todas las que siguen en T_2 por las que siguen en T_1 o repetir el mismo procedimiento tomando como arista inicial a e_{i+1} y de esta manera llegamos a que T_2 no era AGM. Luego, el AGM es único.

↳ mmm.. SE PUEDE EXPLICAR MÁS PERO OK.

Tomo G /

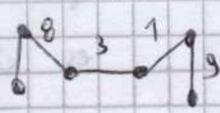
$AGM(G) =$



$W(AGM(G)) = 20$

Sean T_1, T_2 segundos mejores AGMs de G:

T_1



$W(T_1) = 23$

T_2

$W(T_2) = 23$

$T_1 \neq T_2$

Luego, el ~~AG~~ segundo mejor AG no es único.

B

5
165

(5) Para modelar el problema tomamos a las esquinas como vértices del grafo y a las calles (que conectan esquinas/vértice) como las aristas con peso el tiempo necesario para recorrer ~~cada calle~~ la calle que representa. Las aristas son direccionadas ~~se~~ acorde a la mano de la calle que representa.
(Suponemos grafo sobre lista de adyacencias)

Idea del algoritmo: aplicar Dijkstra desde el depósito a ~~todos los otros~~ todos los destinos con complejidad $O(\log(V) \cdot V + E \cdot V)$,
~~m~~ invertir las direcciones de las aristas para convertir los caminos del destino al depósito en caminos del depósito al destino (probado ~~en~~ en clase)
~~[illegible]~~
aplicar Dijkstra nuevamente del ~~destino~~ depósito a todos los destinos. Devolver la suma de los tiempos de los caminos mínimos del depósito a los destinos en el primer grafo ~~[illegible]~~ más la suma de los caminos mínimos del depósito a los destinos (igual al ~~la~~ camino mínimo ~~del de~~ los destinos al depósito) en el grafo con las direcciones invertidas.

(se puede hacer mejor directamente en la lista)

Puedo invertir las direcciones de las aristas pasando la lista de adyacencias a matriz de adyacencias en tiempo $O(n^2)$, invertir las direcciones en la matriz de adyacencias haciendo ~~[illegible]~~ swap(M[i][j], M[j][i]) $\forall i, j \in \{1, \dots, n\}$ ~~[illegible]~~ con compl. total $O(n^2)$.

Después puedo volver a la lista de ady. ~~[illegible]~~ nuevamente en $O(n^2)$.

De esa manera, en $O(n^2)$ tengo el grafo ~~[illegible]~~ con las direcciones invertidas.

Le aplico Dijkstra del ~~[illegible]~~ depósito a todos en ambos grafos, devolviendo en cada caso un diccionario de distancias. implementado sobre vector

Esto se hace ~~[illegible]~~ con complejidad ~~[illegible]~~ $O(2(\log(n) n + m))$ = ~~[illegible]~~ $O(\log(n) n + m)$

Por último, recorro ~~el~~ ~~[illegible]~~ ambos diccionarios sumando todo en cada uno y sumando los resultados entre sí.

Esto se logra con complejidad $O(n)$.

Como m en peor caso es $\frac{n(n-1)}{2}$, $m \in O(n^2)$.

Luego, podemos tomar que Dijkstra tiene complejidad $O(n^2)$.

La complejidad total del algoritmo queda $\underbrace{O(n^2)}_{\text{obtener el grafo con direcciones invertidas}} + \underbrace{O(n^2)}_{\text{caminos mínimos}} + \underbrace{O(n)}_{\text{sumar tiempos}}$

$= O(n^2)$ ✓

Veamos que invertir las direcciones preserva los caminos mínimos. Sea $C = v_1 v_2 \dots v_k$ un camino mínimo de v_1 a v_k en un digrafo G con $v_i \in V(G)$.
Sea G' el digrafo G con las direcciones invertidas. $C' \in G'$ y $C' = v_k v_{k-1} \dots v_2 v_1$ con la suma de los pesos de las aristas igual a la de C. Como podemos hacer lo mismo para cualquier camino de G, el camino mínimo en ~~es el mismo con las~~ G' de un nodo v_j a v_i es el camino mínimo de v_i a v_j en G invirtiendo el sentido de las aristas.